



US 20080293474A1

(19) **United States**

(12) **Patent Application Publication**
Douceur et al.

(10) **Pub. No.: US 2008/0293474 A1**

(43) **Pub. Date: Nov. 27, 2008**

(54) **MOTIVATIONAL DEPLOYMENT
MECHANISM FOR NETWORKED SYSTEMS**

Related U.S. Application Data

(60) Provisional application No. 60/940,001, filed on May 24, 2007.

(75) Inventors: **John R. Douceur**, Bellevue, WA (US); **Thomas Moscibroda**, Redmond, WA (US)

Publication Classification

(51) **Int. Cl.**
A63F 9/24 (2006.01)
(52) **U.S. Cl.** **463/17**
(57) **ABSTRACT**

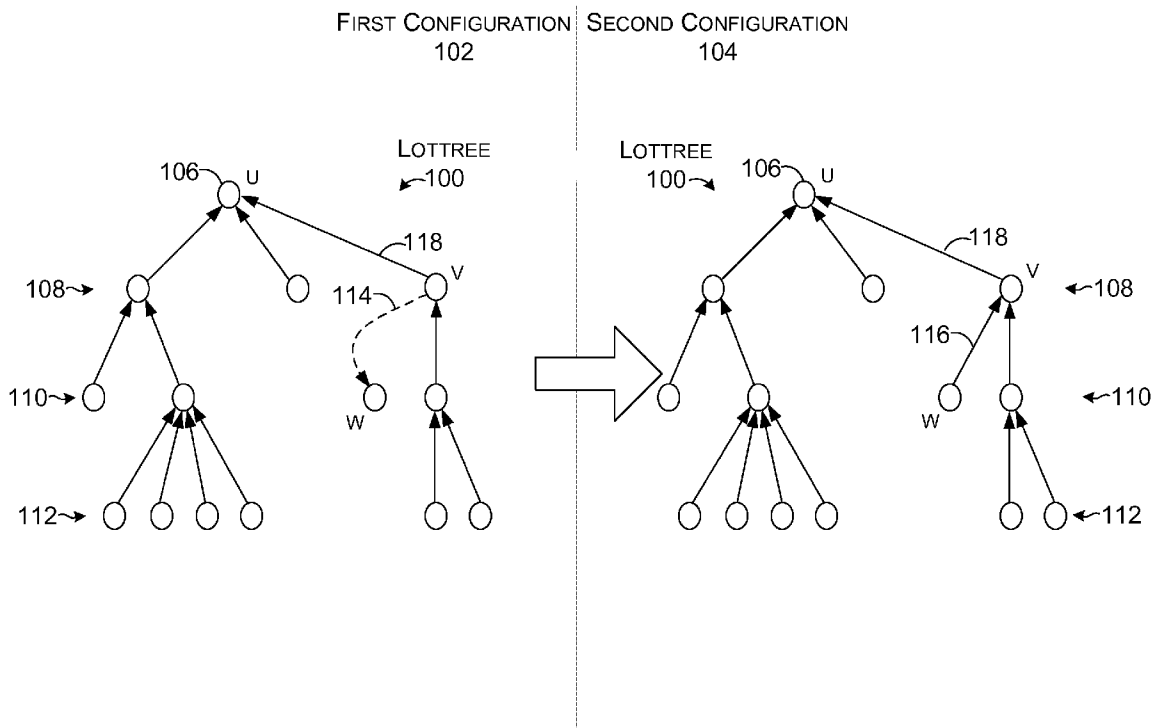
Correspondence Address:
LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

The described implementations relate to networked or distributed systems and more particularly to providing motivation for deployment of networked systems. One technique gathers a solicitation hierarchy in a distributed system. This technique also tracks contribution to the distributed system of participants within the hierarchy. This technique further probabilistically determines a participant as a lottery winner based at least in part on the solicitation hierarchy and the contribution.

(73) Assignee: **Microsoft Corporation**, Redmond, WA (US)

(21) Appl. No.: **11/971,541**

(22) Filed: **Jan. 9, 2008**



FIRST CONFIGURATION SECOND CONFIGURATION
102 104

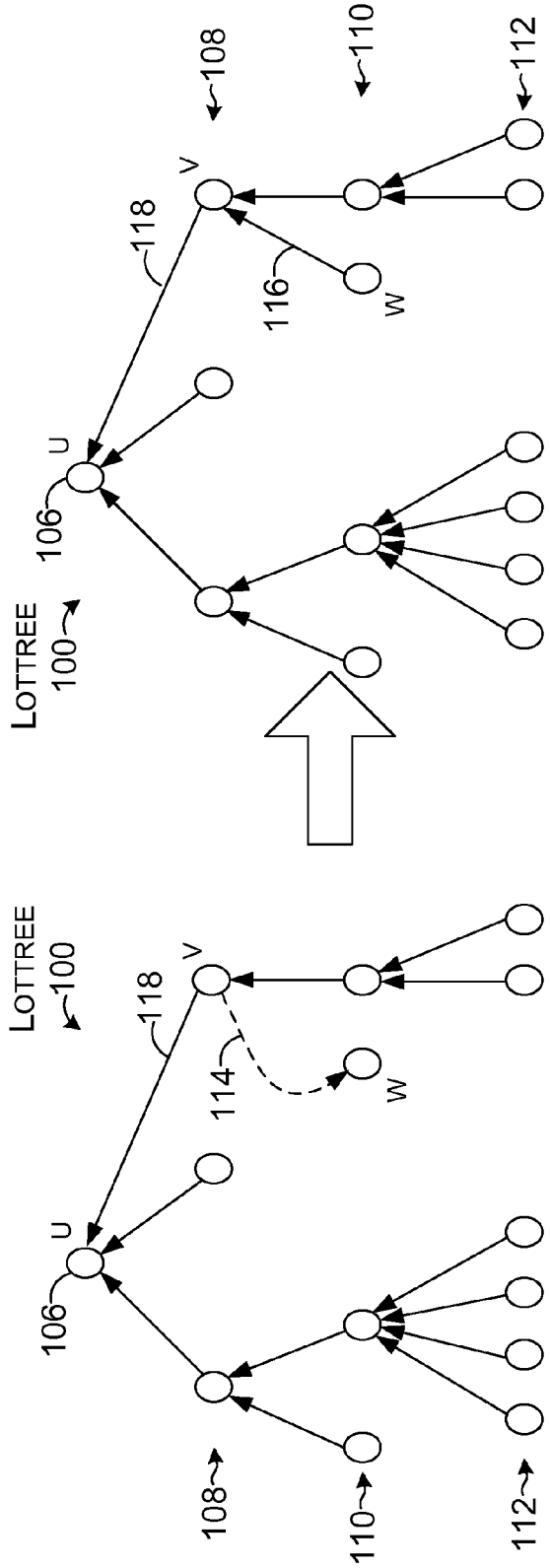


FIG. 1

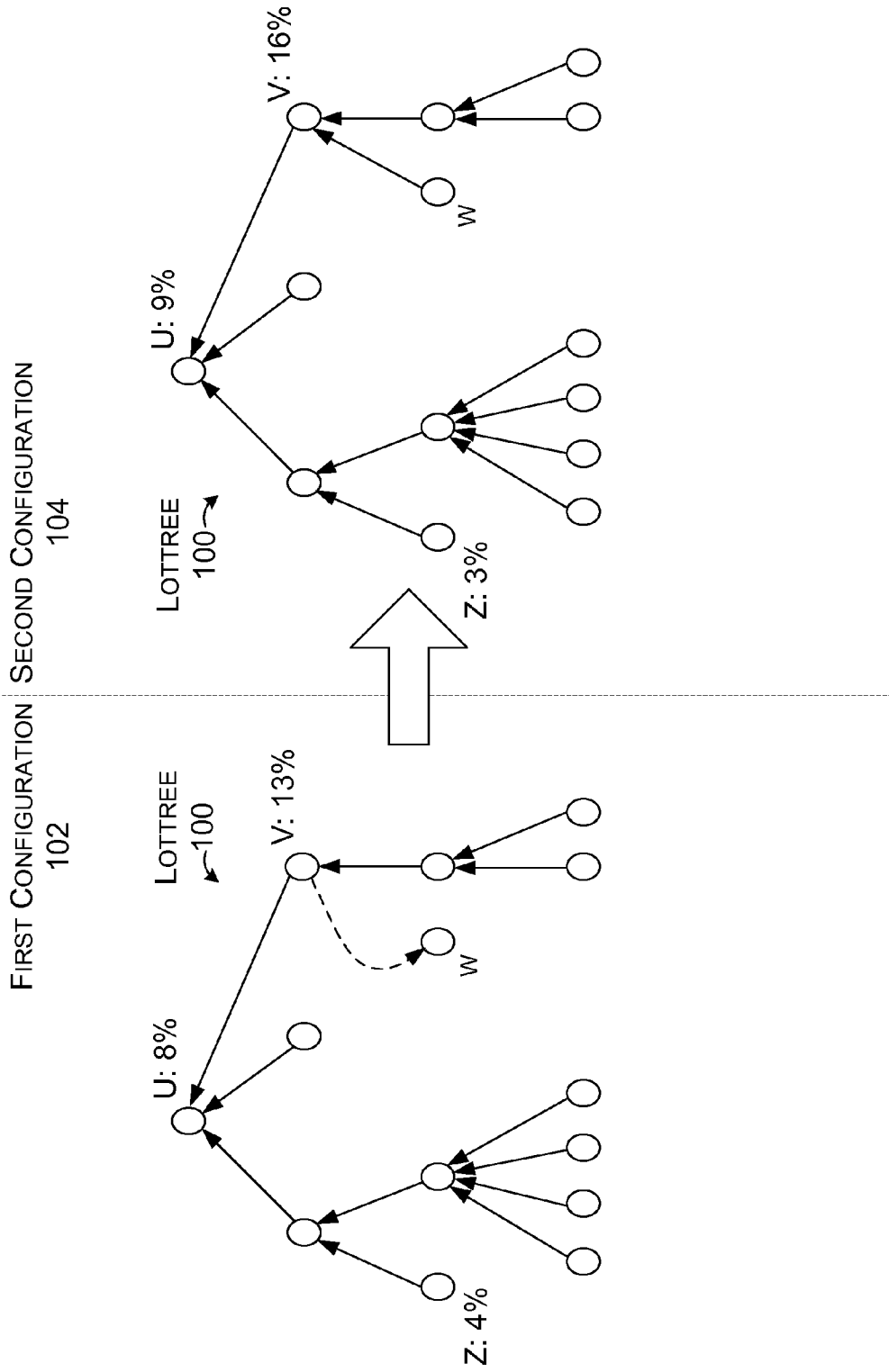


FIG. 2

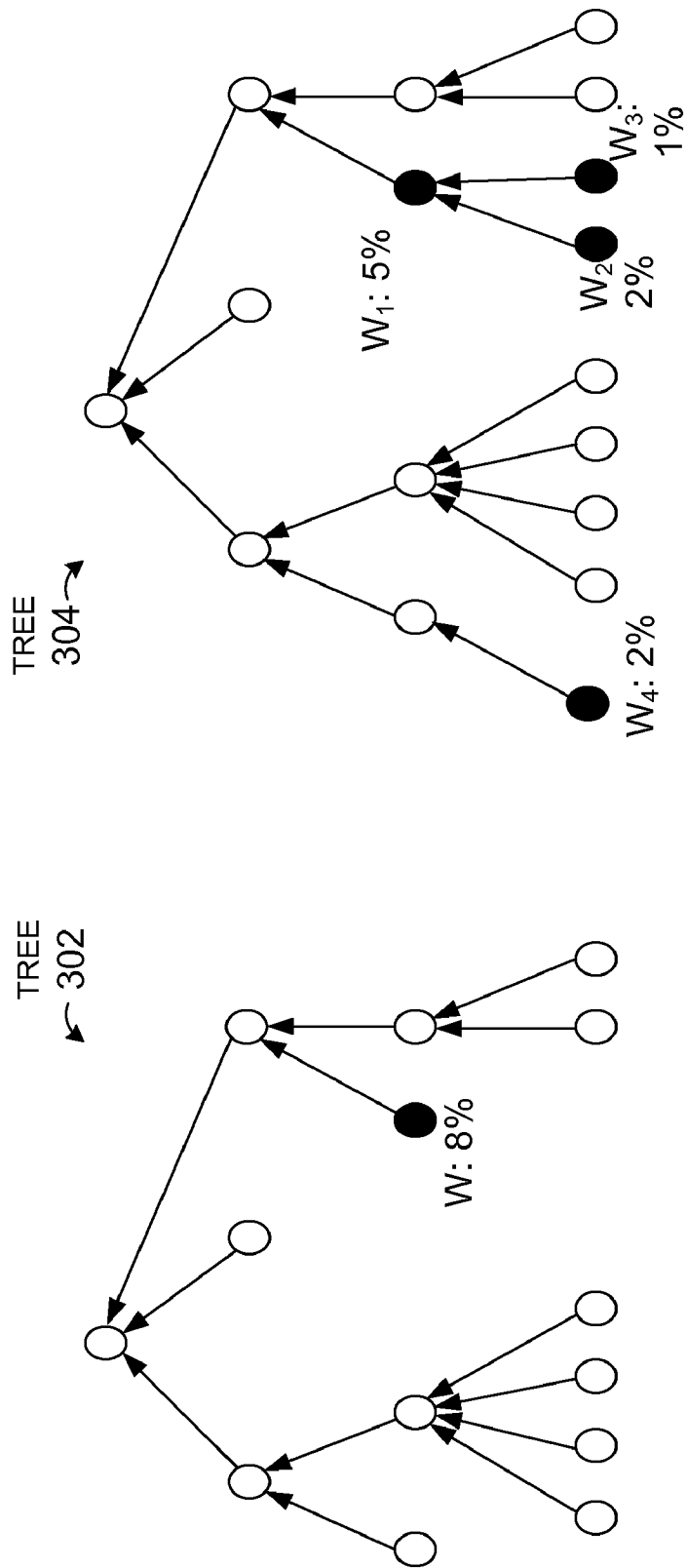


FIG. 3

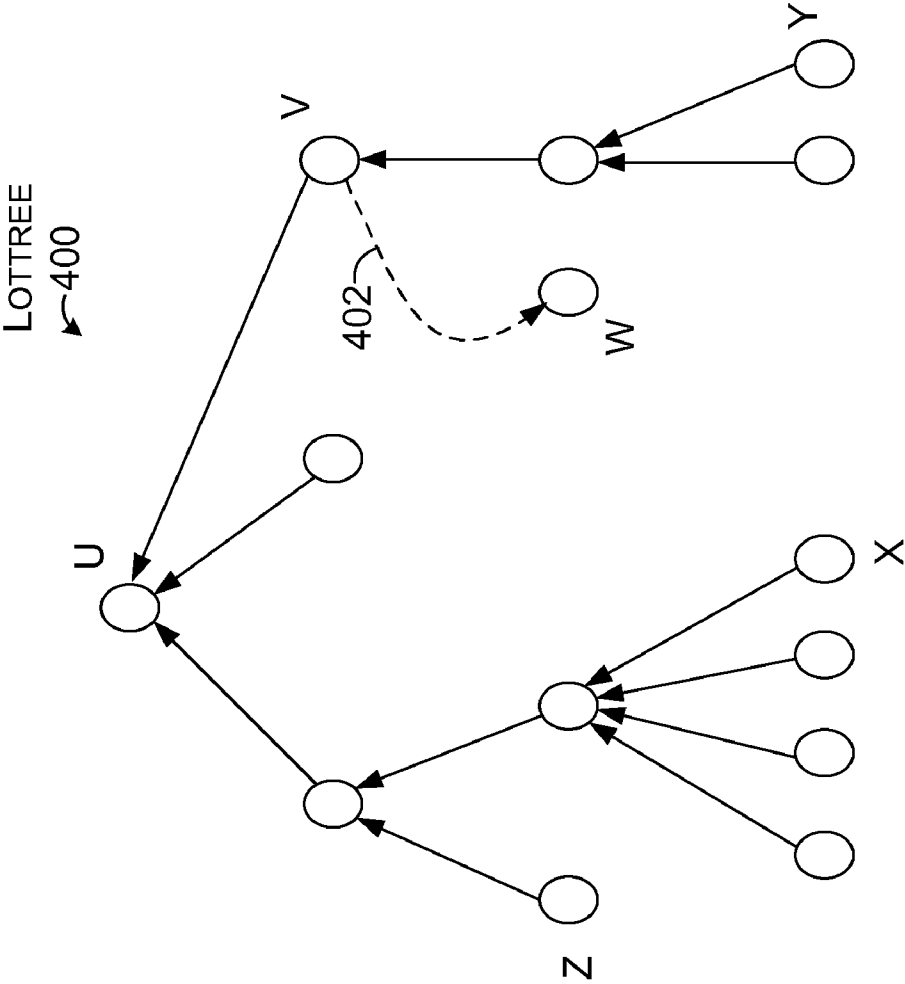


FIG. 4

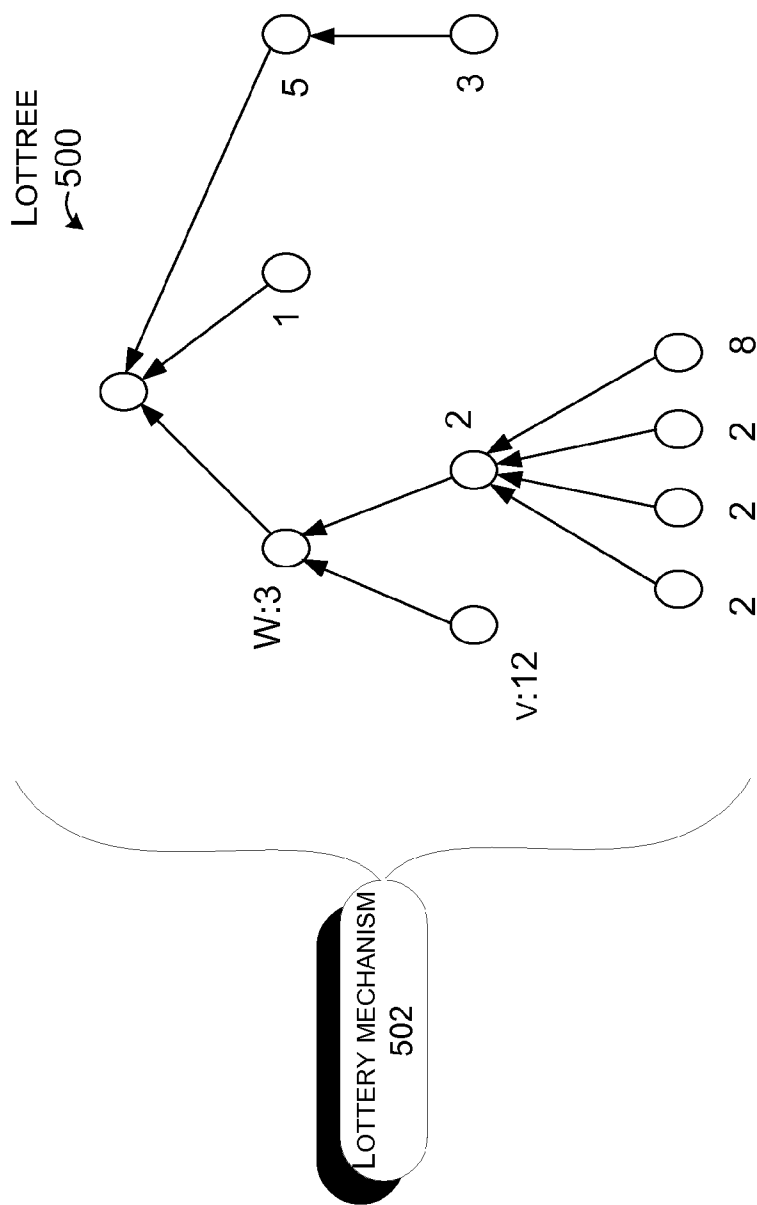


FIG. 5

Luxor Tree Winner Selection 600
(without rescaling)

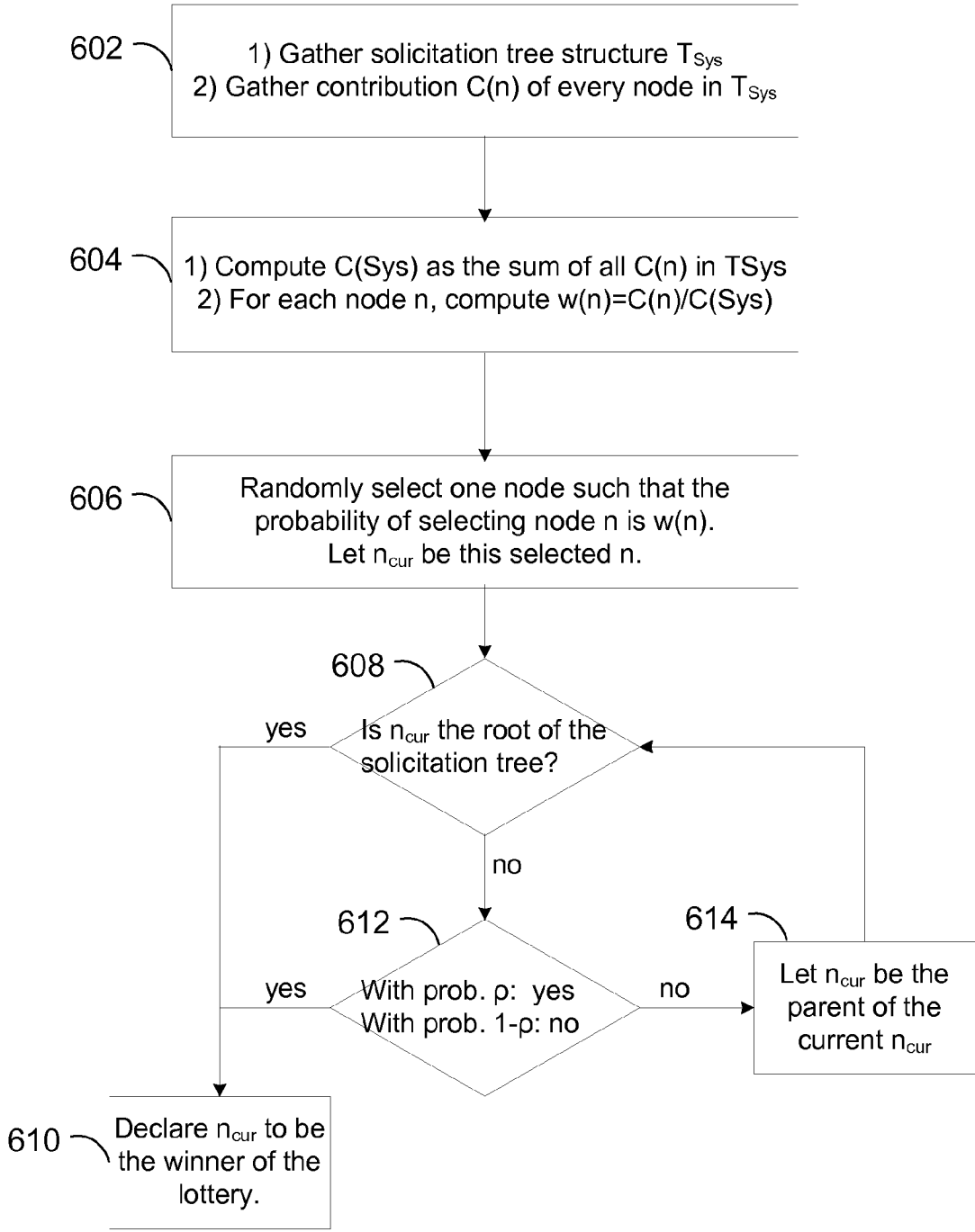


FIG. 6

Pachira Tree Winner Selection 700
(without rescaling)

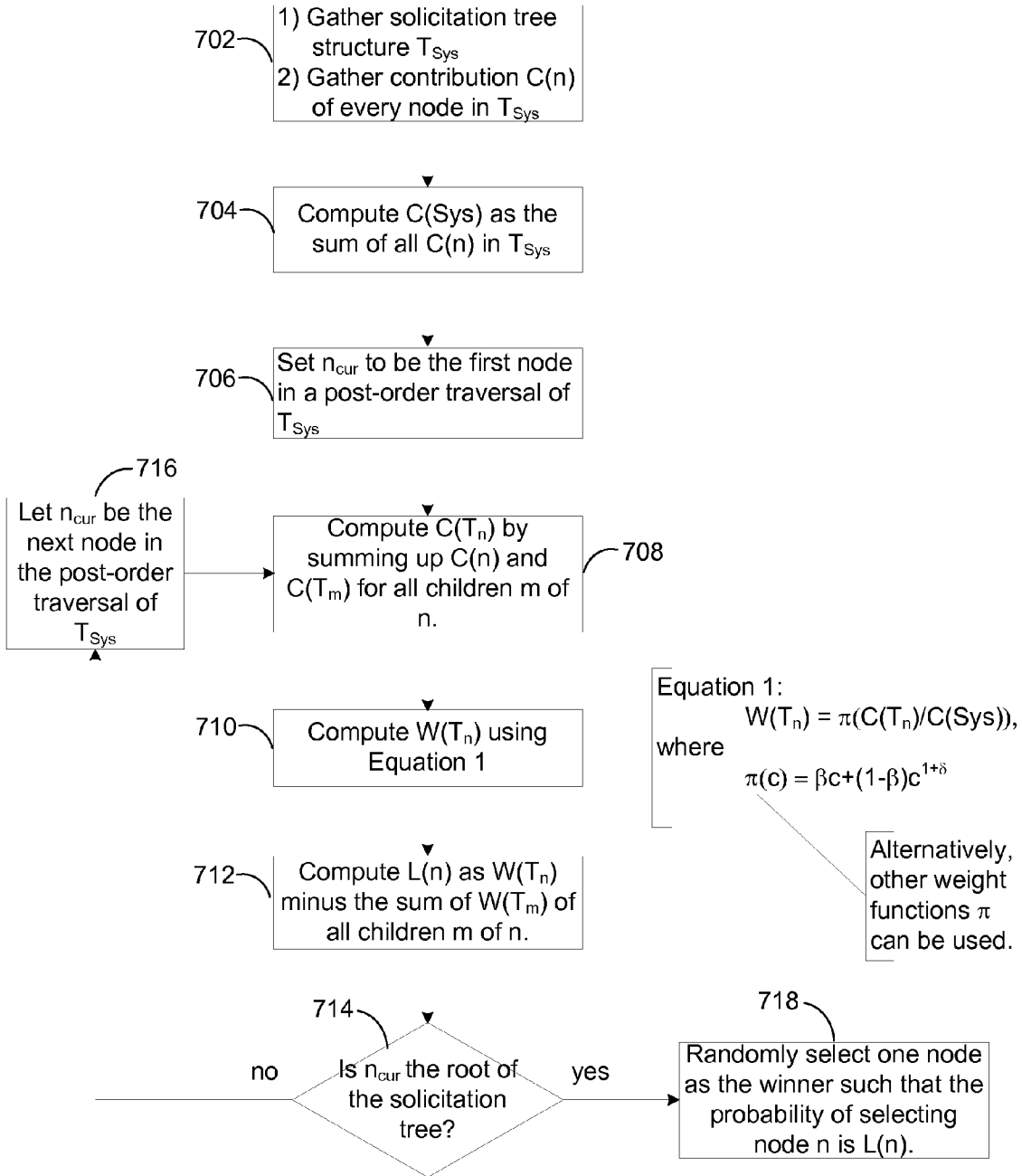


FIG. 7

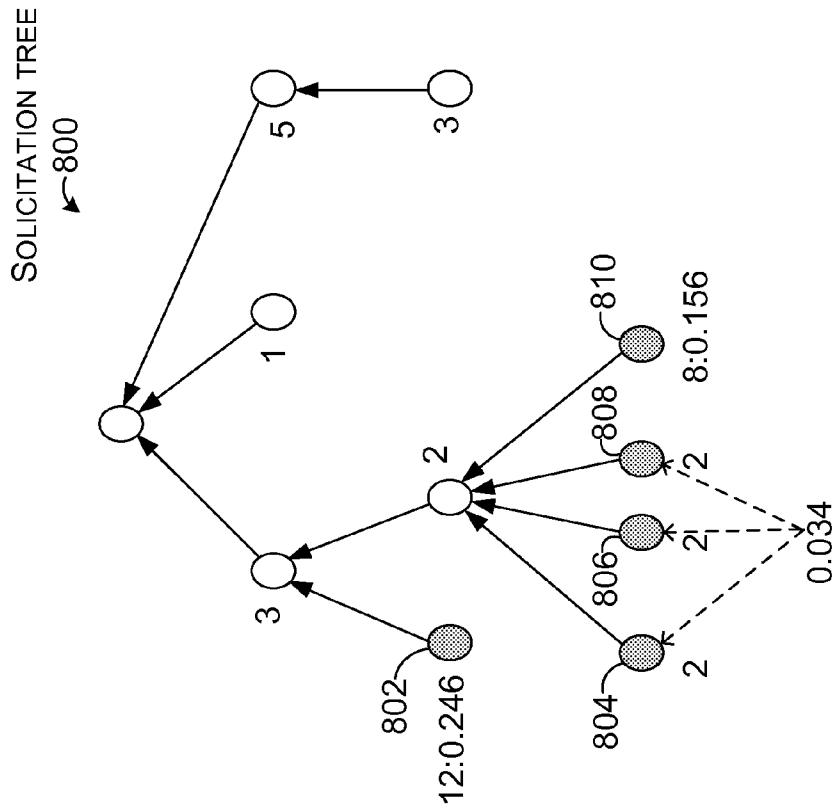
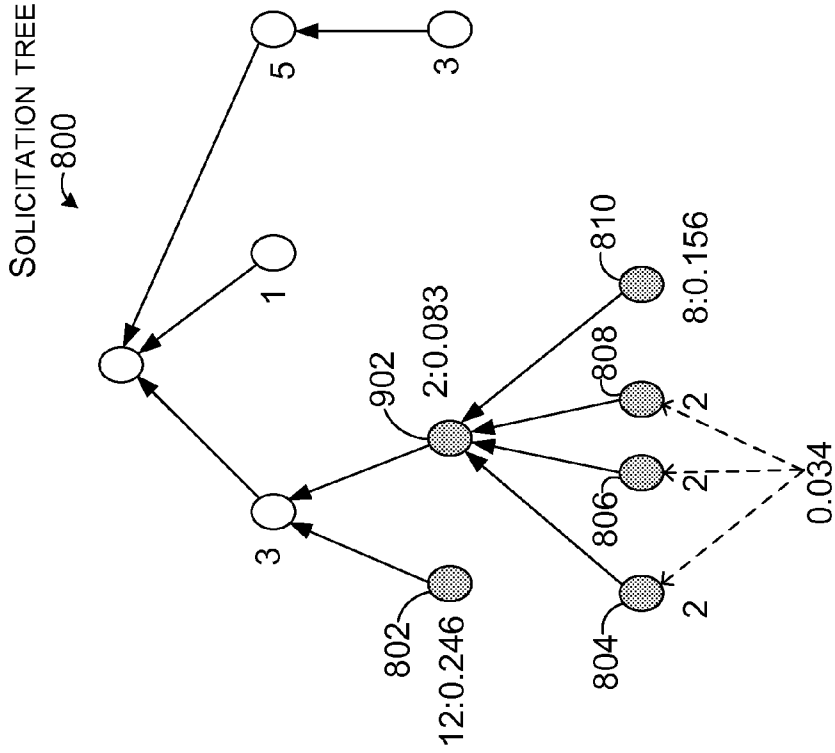


FIG. 8

FIG. 9

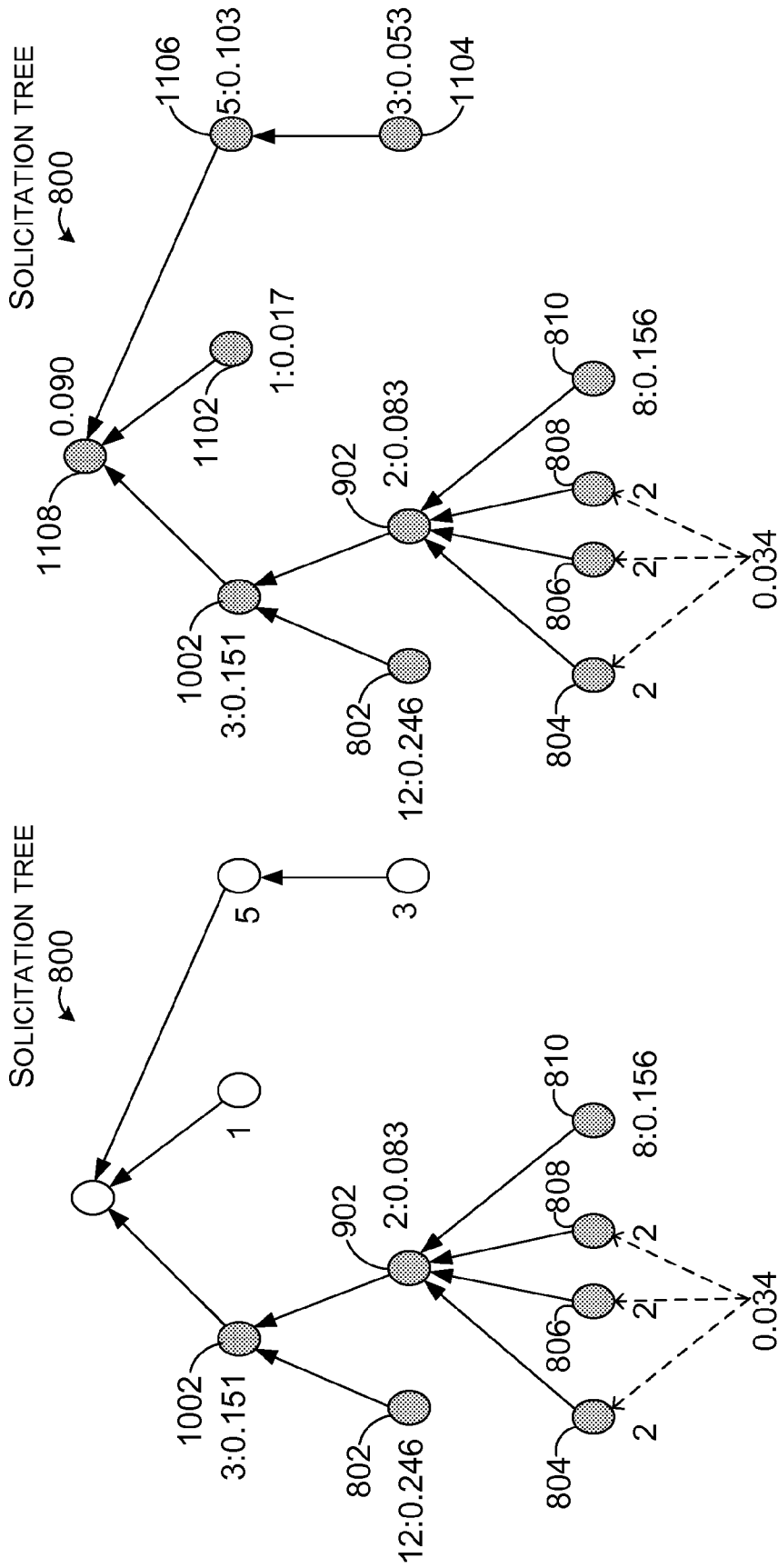


FIG. 10

FIG. 11

**MOTIVATIONAL DEPLOYMENT
MECHANISM FOR NETWORKED SYSTEMS**

PRIORITY

[0001] The present patent application claims priority to U.S. Provisional Patent Application No. 60/940,001, filed May 24, 2007, and titled "Motivational Deployment Mechanism for Networked Systems". The disclosure of the above mentioned provisional patent application is herein incorporated by reference in its entirety.

BACKGROUND

[0002] The present application relates to networked systems.

SUMMARY

[0003] The described implementations relate to networked or distributed systems and more particularly to providing motivation for deployment of networked systems. One technique gathers a solicitation hierarchy in a distributed system. This technique also tracks contribution to the distributed system of participants within the hierarchy. This technique further probabilistically determines a participant as a lottery winner based at least in part on the solicitation hierarchy and the contribution.

[0004] Another network system includes a lottery tree mechanism to encourage one or more of: network participation and network solicitation. The above listed examples are intended to provide a quick reference to aid the reader and are not intended to define the scope of the concepts described herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The accompanying drawings illustrate implementations of the concepts conveyed in the present application. Features of the illustrated implementations can be more readily understood by reference to the following description taken in conjunction with the accompanying drawings. Like reference numbers in the various drawings are used wherever feasible to indicate like elements. Further, the left-most numeral of each reference number conveys the Figure and associated discussion where the reference number is first introduced.

[0006] FIGS. 1-5 illustrate exemplary lottery trees that can be employed to provide motivation for deployment of networked systems in accordance with some implementations of the present concepts.

[0007] FIGS. 6-7 are flow diagrams of exemplary techniques relating to motivating deployment of networked systems in accordance with some implementations of the present concepts.

[0008] FIGS. 8-11 illustrate exemplary lottery trees that can be employed when providing motivation for deployment of networked systems in accordance with some implementations of the present concepts.

DETAILED DESCRIPTION

Overview

[0009] This patent application describes lottery trees (hereinafter, "lottrees"), i.e., motivational deployment mechanisms for networked systems. Included in the application are examples for using sweepstakes for motivating participation

and solicitation in networked systems. In individual sweepstake examples, odds associated with the sweepstake can depend on a contribution and solicitation structure. Multiple exemplary sweepstake schemata that include these concepts are described below.

Introduction

[0010] Many of the most interesting networked systems that have been proposed in the research literature in recent years are intended to be deployed on end hosts, which are under the control of individuals. Some of these systems are asymmetric, in the sense that the participants contribute resources or effort to the system but receive nothing directly in return. Other systems, although symmetric insofar as the contributors are also the benefactors, rely on the network effect to make the benefit of the system significant. A problem that can prevent a large number of such networked systems from ever becoming popular is bootstrapping, i.e., attracting a sufficiently large initial user base to guarantee sustainable growth of the networked system.

Motivational Deployment in Symmetric/Asymmetric Systems

[0011] Finding a way to motivate users to participate in a networked system can be important if the system itself does not provide enough immediate benefit to the user. Two scenarios are described below in this regard.

[0012] Symmetric network-effect systems, such as file-sharing services, recommendation networks, social forums, open databases, or collaborative references works, can become self-sustaining when the scale becomes large enough for the benefit of participation to outweigh the cost. However, such systems are notoriously difficult to bootstrap, as evidenced by the numerous developed peer-to-peer systems, few of which have become popular. In such symmetric network-effect systems, a user typically benefits from participating in the system only if a large number of other users also participate. The problem is that if a network effect is the main feature attracting users to participate in a system, it is difficult for a system that does not yet have a sufficiently large user base to get new participants. This bootstrapping problem is sometimes referred to as the Cold Start Problem.

[0013] Asymmetric distributed systems, such as BOINC, Folding@Home/Genome@Home, and GPU, are potentially even more problematic. Potential contributors are asked to provide spare resources such as computing cycles, storage, or bandwidth toward a goal that does not directly benefit them. Therefore, the potential contributors have little or no incentive to participate in the system. Evidently, some people do choose to contribute, for various reasons including a selfless desire to help, a hope that the work may eventually benefit them, the "geek chic" associated with high contribution levels displayed on public ranking sites, and even the meager value of looking at pretty pictures on a screensaver. Once such systems reach a threshold of popularity, they seem able to sustain substantial ongoing contribution. Following the principle of "a crowd draws a crowd", the media attention and buzz that accompanies a large congregation can inspire others to join. In most cases, however, potentially useful systems languish in unpopularity, having never managed to inspire a critical mass of participants. For instance, of the 36 active BOINC projects, SETI@home has gathered 65% of the contributing users and 80% of the contributing machines.

[0014] Bootstrapping such systems and motivating users to participate in them leads to two challenges. First, participants might reasonably expect their investment of effort and resources to return some palpable value, which neither asymmetric systems nor small network-effect systems provide. The simple expedient of monetarily compensating early adopters is often not a practical option, if, for instance, the system is deployed by a small research group whose limited budgets may be vastly insufficient to compensate contributors at a level that many would find satisfactory. That is, there is not enough participation incentive.

[0015] Second, participants have little or no incentive to persuade their friends and acquaintances to join. Even for network-effect systems, wherein the value of the system grows as the population grows, the marginal benefit provided by each new participant is diffusely spread among the entire pool of participants, rather than accruing significantly to the person who solicited the new member. Thus, there is no inherent solicitation incentive that fosters system propagation.

[0016] The present concepts address these two challenges with a general mechanism for motivating bootstrap deployment of networked systems. The mechanism or motivational deployment mechanism, herein called lottery trees (lottrees), employs the leverage of lottery psychology to disproportionately motivate people to contribute to a developing system. In addition, lottrees employ a mechanism similar to a multilevel marketing scheme to motivate participants to solicit other people to contribute as well. Consequently, lottrees can significantly increase the rate of network deployment and/or reduce the financial investment required to ensure rapid and eventually self-sustaining growth.

[0017] The present concepts can motivate contribution by means of a recurring sweepstakes, which probabilistically rewards people for participating in the system (installing software, providing bandwidth, submitting recommendations, actively participating in a social network, etc . . .), and also for encouraging their friends and associates to participate as well. In one instance, the basic idea of lottrees is that nodes accrue points when actively participating in the networked system. In addition, contributors can send their friends a solicitation to join the networked system, and these referral relationships can provide bonus points to the referrers. In at least some instances, lottrees can be devised and/or applied in such a way that participation incentive, solicitation incentive, and fair-play is ensured.

EXAMPLES

[0018] There are numerous examples where lottrees can be helpful to boost the number of participants in a networked system, particularly in its bootstrapping phase. A first example can relate to distributed computing or grid computing projects like SETI@home in which participants are expected to contribute central processing unit (CPU) cycles to a networked system from which they do not benefit directly. A second example can pertain to recommendation networks (hotels, books, CDs, . . .). Such networked systems can provide a lot of benefit to users once there are already a lot of recommendations available. Lottrees can be used to bootstrap such a networked system. A third example can relate to network-based tagging or classification of pictures. In such a system, no participant has a direct benefit of participating, but lottrees can be used to motivate participation. A fourth example can involve a distributed, decentralized file-storage

system in which participants are expected to offer available disk memory to the system. Lottrees can similarly be applied to peer-to-peer (P2P) applications. A fifth example can relate to online gaming sites. Again, the network effect can be crucial here, especially for massively multiplayer games. In many instances, such games are attractive only if there are already a sufficient number of existing players. Lottrees can be used to achieve a bootstrapping of such systems. A sixth example can relate to online social networks, where lottrees can help in attracting new users to participate in the social network. In summary, lottrees can be employed with any system that has a capability or mechanism for tracking solicitation (i.e., who solicited whom) and/or a capability or mechanism for tracking contributions of individual participants. An individual's probability of winning can then be based, at least in part, on one or both of the tracked solicitations and tracked contributions. The skilled artisan should recognize numerous other applications of lottrees that utilize the concepts described above and below.

[0019] Existing Approaches

[0020] No incentive scheme currently exists for networked systems where participants do not have any (or not enough) direct benefit because it is either an asymmetric network system or a symmetric network system that has not yet reached a sufficient size to make use of the network effect. A relatively simple way of acquiring a high participation for such a networked system would be to individually pay (by money or the distribution of other items of value) each participant. In other words, participants can be paid for joining the network if the network itself does not provide enough incentive for users to join. A potential problem with this approach (i.e., direct payment) is that it may not scale. Direct payment can require the investment of large budgets, which in many cases may be unavailable. Another difficulty of individually paying each participant is the overhead involved in such a process. In contrast, exemplary lottrees can provide a way of recruiting high participation even with a limited budget.

[0021] A potentially cheap alternative to paying each participant individually is to do nothing and simply hope that there are enough "benevolent" participants and early-adopters that join the network, although they do not benefit from doing so. While success stories like this exist, most networked systems (in the research community as well as in business situations) tend to fail because the networked systems cannot acquire a sufficiently large user base to overcome the bootstrapping problem.

[0022] In symmetric systems, such as the P2P BitTorrent system, tit-for-tat-based incentive mechanisms have been widely employed and studied. The premise is that every user can essentially profit from participating in the network and such incentive mechanisms merely prevent some users from selfishly using the system without also contributing to it. However, such mechanisms do not help in situations in which users do not have a specific benefit for joining in the first place. Motivational deployment generated by lottrees can therefore be fundamentally different from such existing tit-for-tat based incentive mechanisms or other mechanisms that have been proposed in the research literature. Again, such mechanisms operate on the premise that people participate in a networked system if the utility they receive from the system is higher than the cost of joining the system. Hence, lottrees

can inherently provide bootstrapping for asymmetric or small-sized systems from which the users would otherwise get little or nothing in return.

[0023] Exemplary Lottrees

[0024] In at least some cases a lottree is a mechanism to provide an efficient way to leverage a small budget into a comparatively large amount of participation. Lottrees can employ a lottery or sweepstake to probabilistically compensate people who participate in a networked system and/or who successfully encourage others to join the networked system as well and/or contribute to it. Depending on the specific networked system under consideration, contributing to or participating in this system can mean such different things as performing computation, storing information, transmitting data, testing a software application, providing recommendations, actively engaging in a social network, and so forth. In summary, contribution can be any effort or expenditure of resources on the part of a participant that either directly or indirectly benefits the system.

[0025] Regardless of the specific nature of the contributions, lottree schemes can work as follows. Assume that there is an executive entity (a person, company, or research group) whose goal is to deploy a networked system for which the entity needs to attract a large number of participants with sufficiently high contribution. Further, assume that this executive entity of a network is willing and able to invest a certain amount of money (or any other item of value or perceived value)—termed the payout—for attracting a sufficient user-base of this network. The payout can be configured to have value or perceived value from the perspective of participants and/or potential participants. For instance, a payout can be in the form of ‘points’ that do not have any actual value, but are perceived as valuable by the participants and/or potential participants. The function of the lottree can be then, after a certain amount of time has passed, to select one contributor of the network as the recipient of the payout. Alternatively, a lottree may periodically select a winner or may opt for choosing multiple winners in each period. The above mentioned mechanisms (among others) fall into the realm of possible lottree strategies. In at least some implementations, the lottree performs this selection in such a way that encourages high contribution, participation, and solicitation among participants.

[0026] For purposes of explanation, consider the above mentioned networked system to be initialized with a single root node which represents the executive entity. Whenever a new person joins the network, he/she does so as a child of some person that is already a node in the system. For example, people might sign up their computers to the network by visiting a web site that records information or installs an application. If someone visits the site on his own, his computer joins as a child of the root. Once a member, he is able to send solicitations, perhaps in the form of coded email links, to friends and associates. Anyone who follows the coded link to the web site will join as a child of the member who sent the link. That way, the solicitation structure (who solicited whom) induces a tree whose root corresponds to the central entity of the networked system, and in which every node represents one participant in the system. The skilled artisan should recognize that other solicitation structures are possible, such as directed graph structures which admit multiple solicitors per joining node.

[0027] FIG. 1 offers a lottree configuration that is consistent with the above example. In this case, FIG. 1 shows a

directed lottree **100** in a first lottree configuration **102** on the left side of the Figure and a second or subsequent lottree configuration **104** on the right side of the Figure. In this example, lottree **100** includes a root node ‘u’ generally at **106** with children nodes represented horizontally across the FIG. at **108**. Subsequent descendant nodes (i.e., grandchildren and great-grandchildren) are shown horizontally across the FIG. at **110** and **112**, respectively. In first lottree configuration **102**, a child node or participant ‘v’ solicits a new participant ‘w’ for the tree as indicated by dashed line **114**. In the ensuing solicitation tree structure configuration **104**, node w is added to lottree **100** as a child of v as indicated by solid line **116**.

[0028] In practice, solicitations may occur in numerous ways. For instance, in one example, participant v can send a recommendation email to friend w containing a link that leads to a sign-up page. When clicking on this link and signing up as a new participant of the networked system, the URL indicates v as the solicitor of w. In another case, at the time of joining the network, a new participant may indicate its solicitor by providing its (e.g., the solicitor’s) name or identification number (e.g., systems with off-line sign-up procedure). For instance, in this case, w would list v as w’s solicitor when completing a sign-up procedure. In still another example, depending on the nature of the networked system, it is also conceivable that the solicitor of a specific node changes during the course of the system’s lifetime.

[0029] Besides the solicitation tree, the second informational aspect maintained about each participant can be its contribution (e.g., the individual participant’s contribution to the networked system). As pointed out above, depending on the networked system for which the lottree is employed, this contribution can for instance be measured utilizing one or more criteria. An example of such criteria can include an amount of computing time (e.g. in CPU cycles) offered to a distributed computing or grid computing project such as SETI@home. Another example can be the number of software updates/products installed from a software company. A further example can be the number of recommendations submitted to a networked recommendation system. Another example can be the number of pictures tagged with keywords in a networked system that classifies picture-databases. A further example can be the amount of storage provided to a distributed, decentralized file-storage system. Still another example can be the number of games played on an online game site or in a massive multiplayer games setting. A still further example can be the number of files or blocks uploaded to a P2P system. The skilled artisan should recognize still other examples.

[0030] In a networked system without any such measurable contribution or when soliciting new participants also coincides with the contribution (as in certain marketing schemes or social networks), the contribution can also be set to be uniform (e.g., equal to 1) over all participants in the lottree.

[0031] When the networked system has grown to a size that the executive entity judges to be sufficient, the executive entity can farm out work units to the participants (nodes in the tree) and can record each node’s contribution. Alternatively, the node’s contributions can be gathered by (or reported to) the executive entity periodically. The lottree can then select a winner based on the tree structure and on contributions of individual nodes. This winner selection can be random, but can prioritize participants with high contribution and many solicited siblings. The winner(s) of the lottree can receive the

payout and a winner may be selected periodically. Winner selection is described in more detail below in the properties section.

[0032] Formal Model

[0033] As discussed above, individual participants that join a lottree can be modeled as nodes in a tree, such that their solicitation history can be tracked. The solicitation structure then induces a solicitation tree, in which a directed edge (i.e., solid line such as solid line **118** of FIG. **1**) from a node v to node u indicates that u was v 's solicitor. A potentially crucial ingredient of lottrees is that, at least in some manifestations, every participant has a certain amount of measurable contribution. As mentioned above, the measurable contribution can, for example, be the number of product reviews submitted to the system, the amount of bandwidth allocated to a P2P system, and/or the number of experiments executed on a local machine. Regardless what exactly constitutes this contribution in a specific system, exemplary implementations can formally express the contribution as a number that is larger if the contribution to the system gets larger.

[0034] Although different lottrees may differ in both functionality and implementation, often they have in common that they select one or more lottery winners based on 1) the topology of the tree (solicitations) and 2) the contribution by individual participants. Hence, based on these commonalities, a lottree can be formalized as a function that, based on the solicitation tree and all the contributions of the nodes, determines the likelihood or probability of a node's winning the sweepstakes or lottery. The lottree can then randomly select a winner according to this probability distribution.

[0035] Properties

[0036] One potential challenge in designing a lottree scheme can be how to define the rules of selecting a winner in such a way that encourages both contribution and system growth. Simple schemes that readily provide some benefits may fail to provide others. For example, one scheme is to employ a simple lottery that randomly selects a winner in proportion to its contribution to the network. Although this encourages contribution (because higher contribution correlates to higher probability of winning), it discourages participants from soliciting others, since any new member can decrease the current members' chances of winning. A specific instance of a lottree mechanism should simultaneously encourage contribution, solicitation, and fair play.

[0037] The following discussion further describes some of these desirable properties that a lottree should satisfy.

[0038] Contribution-Incentive and Fairness

[0039] The lottree scheme should be set up in such a way as to encourage contribution from the participants that have already joined. This implies, for instance, that a participant's odds of winning should not decrease when the participant increases its contribution. Also, a lottree should enforce some kind of fairness among all participants. For example, if a participant contributes 1% of the total system contribution, that participant's odds of winning should not be significantly below 1%. Maintaining such a notion of fairness motivates high contribution, and it also ensures that even when the tree has already grown to a considerable size, there is motivation to join the tree as a participant.

[0040] Solicitation-Incentive

[0041] The main purpose of lottery trees is to foster the growth of a networked system and hence, it is beneficial that its winner selection mechanism offers as much incentive for soliciting new participants as possible. Consider, for

example, a simple lottery that selects the sweepstakes winner simply based on the contributions and regardless of the solicitation structure. Although this is a valid lottery scheme, it does not encourage solicitation. In particular, no participant has an incentive to attract new participants because every such new participant decreases the earlier participant's probability of winning.

[0042] Instead, at least in some instances, a good lottree scheme should ensure that whenever a participant solicits a new participant to the system, its (i.e., the soliciting participant's) probability of winning (or expected payout) increases. In this case, every participant will have an incentive to acquire as many new children as possible, which results in rapid growth of the overall system.

[0043] FIG. **2** builds upon the above description of FIG. **1** in relation to lottree **100** by illustrating expected gain relative to recruitment. In a similar fashion to FIG. **1**, FIG. **2** shows lottree **100** in a first configuration **102** before a solicitation and in a second configuration **104** after the solicitation. In FIG. **2**, the expected gained payout (probability of winning the sweepstakes) of node v should increase when soliciting w as a new child. In the example, the winning probability of node v increases from 13% to 16% by acquiring w as its new child. Specifically, the odds of winning for node v in the first configuration **102** equals 13%. In contrast, v 's odds of winning increases to 16% after recruitment of w in second configuration **104**. Similarly, the winning probability of any ancestor of the new participant may also increase. On the other hand, (because the winning probabilities always sum up to one), the winning probabilities of other nodes (e.g. z) in the tree that are not ancestors of w may decrease. For instance, in this example z 's probability of winning dropped from 4% in the first configuration to 3% in the second configuration as a result of non-related v 's recruitment of w . Utilizing the above mentioned techniques, every participant has an interest in gaining as many new participants as possible.

[0044] Fairplay and Security

[0045] Besides attracting contribution and providing incentives for solicitation, lottrees should also be secure against various notions of strategic behavior of its participants. Particularly, the lottree mechanism should help ensure that participants cannot increase their odds by gaming the system. For one, the mechanism should reduce any potential reward or advantage to participants who join the lottree multiple times and split their contributions among their components. That is, the mechanism should ensure that no participant can gain an advantage (in terms of likelihood of winning or expected payout) by splitting its identity into two or more parts and joining the networked system with these multiple identities. In the context of networked systems, this form of gaming is known as a Sybil attack.

[0046] Consider for example the following two trees **302**, **304** shown in FIG. **3**. In the left tree **302**, participant w has joined the networked system as a single node and the participant's contribution is sufficient to get winning odds of 8%. In the right tree **304**, w games the system and joins the lottree four times as nodes w_1 - w_4 , each time with a different identity, attributing to each identity a part of its overall contribution. In the right tree **304**, the total winning probability of node w (summed up over all its four "Sybil identities" w_1 - w_4) is 10%, which is larger than in the left tree **302**. That is, node w has an incentive to employ a Sybil attack. A good lottree system should avoid this kind of attack by assigning the odds of

winning in such a way that a participant can never be better off by cheating the system in such a way.

[0047] As shown in FIG. 4, another form of gaming the system that a good lottree scheme should prevent is bypassing solicitors. In this case, FIG. 4 shows an exemplary lottree 400. In networked systems, if new participants tend to join the system not as children of the nodes that solicited them, then participants might lose interest in soliciting new participants. Hence, a good lottree scheme should ensure that a newly joining participant never has an incentive to join the tree other than as a child of its solicitor. For example, if a node w receives an invitation to join the system by an existing participant v as indicated at 402, w should never be better off of choosing instead to join the tree as a direct child of the root or any other node. This property should hold regardless of where v is located in the tree. More concretely, after joining the networked system upon solicitation by v, w's probability of winning should never be larger when it joins the tree as a child of any other node than v (e.g. u, x, y, or z).

[0048] The following discussion presents two lottree schemes that exhibit most of these desirable properties and shows how they can be implemented.

Luxor Lottree

[0049] A simple lottree scheme that provides a solicitation incentive is the Luxor lottree scheme. It (i.e., the Luxor lottree scheme) is an extension of the above-described weighted lottery scheme in which each participant's odds of winning corresponds to its relative contribution. The Luxor scheme differs from this simple weighted lottree scheme at least in that each node passes some of its odds of winning up to its parent. That way, participants with many children and descendants have a higher probability of winning the lottery.

[0050] In one form, the Luxor scheme is characterized by two parameters μ and ρ that describe probabilities between 0 and 1, and can proceed in two passes. First, the Luxor scheme can randomly select a participant m in a tree in proportion to its (i.e., m's) contribution. That is, if the total contribution provided by all nodes is 200, and node x has a contribution of 10, then the probability of x being selected as the node m in this first pass is 5%. The node m thus selected is designated as a candidate. Such a candidate wins the lottery with probability μ . With probability $1-\mu$, on the other hand, the winner is one of m's ancestors. Particularly, when m is selected, the candidacy moves to m's parent with probability $1-\mu$. The parent of m is selected as the lottery's winner with probability ρ , and with probability $1-\rho$, the candidacy moves to the current candidate's parent (to m's grandparent), and so forth. In general, the Luxor scheme moves incrementally up the path from m to the root, letting each successive candidate win the lottery with probability ρ . As soon as a candidate is selected as the winner, the process stops.

[0051] The parameter μ can be used to tune the tradeoff between solicitation incentive and fairness. Increasing μ increases fairness at the expense of decreasing solicitation incentive. A practical setting of the parameters could for instance be $\mu=0.7$ and $\rho=0.5$, but other values are conceivable as well.

[0052] Consider the example in FIG. 5 that shows a lottree 500. In this case, the lottree 500 can be generated by a lottery tree mechanism 502 that can then utilize the lottree in selecting a lottery winner(s). In lottree 500, the number at each node indicates this participant's contribution. For instance, the contribution of w is 3 while v's contribution is 12. The total

contribution is 40, which means that a node with contribution x is selected as the initial candidate with probability $x/40$. Assume that the node v is selected (which happens with probability $12/40=3/10$). With probability $\mu=0.7$, v is the winner, but with probability $1-\mu=0.3$, w becomes the candidate. Assume that this latter case occurs. In this case, w is selected to be the winner with probability $\rho=0.5$ and with probability 0.5, the candidacy moves up to the root. In the latter case, the lottery can either be repeated or no prize can be given. The skilled artisan should appreciate that different probabilities could be applied at different levels of the hierarchy; for example, a value ρ_0 could be used for the candidacy to pass up from the parent of the initial candidate, and then a value ρ_1 could be used for the candidacy to pass up from the grandparent of the initial candidate, etc. Furthermore, these probabilities could be based on the distance from the initial candidate, or on the distance from the root, or on one or more combinations of various factors.

[0053] FIG. 6 illustrates a Luxor tree winner selection flow-chart of a method or technique 600. Technique 600 offers an example of the above defined random process of selecting a winner given the solicitation evidenced in the lottree and each participant's contribution. The order in which the technique 600 is described is not intended to be construed as a limitation, and any number of the described blocks can be combined in any order to implement the technique, or an alternate technique. Furthermore, the technique can be implemented in any suitable hardware, software, firmware, or combination thereof such that a computing device can implement the technique. In one case, the technique is stored on a computer-readable storage media as a set of instructions such that execution by a computing device, causes the computing device to perform the technique.

[0054] At block 602, the technique gathers a solicitation tree structure (T_{sys}) and the technique gathers contribution ($C(n)$) of every node in T_{sys} . At block 604, the technique also computes $C(sys)$ as the sum of all contribution $C(n)$ in T_{sys} . For each node n the technique computes $w(n)=C(n)/C(sys)$. In summary, in this case, a lottree can be thought of as a solicitation tree and accompanying contribution data.

[0055] At block 606 the technique randomly selects one node such that the probability of selecting node n is $w(n)$. Let n_{cur} be this selected n. At block 608 the technique queries whether n_{cur} is the root of the solicitation tree. In an instance, where n_{cur} is the root, then the technique can declare the n_{cur} to be the winner of the lottery at 610. In an instance where n_{cur} is not the root, then the technique proceeds to block 612. At block 612 the technique generates a random number in the range [0,1] and queries whether this random number is less than ρ . The technique proceeds to block 610 in an instance of "yes" at 612, which occurs with probability ρ . Conversely, the technique proceeds to block 614 in an instance of "no" at 612, which occurs with probability $1-\rho$. At block 614, the technique proceeds to let n_{cur} be the parent of the current n_{cur} . The technique then returns to block 608 and continues.

Pachira Lottree

[0056] The Luxor tree has the potential disadvantage of allowing certain forms of strategic behavior of participants that can result in an increased probability of their being selected as the winner. In particular, the Luxor tree is vulnerable to Sybil attacks. A Pachira lottree can be shown to exhibit a better behavior in this regard because it is relatively more robust against Sybil attacks.

[0057] The Pachira lottree has two input parameters β and δ that trade off solicitation incentive against fairness. In its most general version, the Pachira lottree is defined using a function $\pi(c)$ defined on $[0,1]$ with the following characteristics:

[0058] $\pi(0)=0$ and $\pi(1)=1$

[0059] The minimum slope of $\pi(c)$ is at least β for all c in $[0,1]$

[0060] The function $\pi(c)$ is convex

[0061] In principle, the Pachira lottree can be defined using any function $\pi(c)$ that follows the above mentioned properties. The following example utilizes a convenient and intuitive function with these characteristics:

$$\pi(c)=\beta c+(1-\beta)c^{1+\delta}$$

where β and $\delta>0$ are the input parameters of Pachira. This scheme makes use of this function in the following way: Each node n in the tree computes its weight $W(n)=\pi(C(n)/C(\text{Sys}))$, i.e., the function $\pi(c)$ applied to the node's proportional contribution. For example, if the sum of all contributions in the tree is 40 and a participant n 's contribution is 5, then n 's proportional contribution is $5/40=1/8$ and its weight $W(n)$ is $W(n)=\pi(1/8)$. In the same way, the weight $W(S)$ of a subtree S in the solicitation tree is defined as $\pi(c)$ applied to this subtree's proportional contribution. For example, the weight of a subtree S consisting of three nodes with contributions 3, 4, and 7 is $W(S)=\pi((3+4+7)/40)=\pi(7/20)$.

[0062] With this definition of weight, the Pachira lottree scheme proceeds as follows. Each node v is assigned an expected value, $L(n)$, defined as the weight of the subtree rooted at n minus the weights of all child subtrees of n .

Implementation

[0063] The Pachira lottree scheme can be implemented and its winner computed in a straightforward way. FIG. 7 illustrates a flowchart of a method or technique 700 that is consistent with one such implementation. The order in which the technique 700 is described is not intended to be construed as a limitation, and any number of the described blocks can be combined in any order to implement the technique, or an alternate technique. Furthermore, the technique can be implemented in any suitable hardware, software, firmware, or combination thereof such that a computing device can implement the technique. In one case, the technique is stored on a computer-readable storage media as a set of instructions such that execution by a computing device, causes the computing device to perform the technique.

[0064] At block 702, the technique gathers solicitation tree structure (T_{sys}). The technique also gathers or tracks contribution $C(n)$ of every node in T_{sys} . At block 704, the technique's algorithm computes the total contribution $C(\text{Sys})$ of the tree by summing up all contributions $C(n)$ of the individual participants $n \in T_{\text{sys}}$.

[0065] Beginning at block 706, this implementation executes a so-called post-order traversal of the tree. That is, all nodes are visited one by one recursively in the order left-most subtree, second left-most subtree, . . . , right-most subtree, node. An example described below in relation to FIGS. 8-11 explains a version of this post-order traversal in more detail. A potentially crucial property of a post-order traversal is that each node is considered only after all results for all its subtrees have already been computed. At each step of this traversal in which a node n is considered, the weight of n 's subtree $W(T_n)$ and the expected value $L(n)$ are computed.

[0066] At block 708 for a node n , this Pachira technique first computes the total contribution $C(T_n)$ of n 's subtree T_n . Note that due to the post-order traversal, all values $C(m)$ for every descendent m of n are already known to the algorithm. At block 710, this Pachira technique computes the weight $W(T_n)$ of n 's subtree by applying the function $\pi(c)$ to the ratio $C(T_n)/C(\text{Sys})$, where $C(\text{Sys})$ denotes the total contribution in the system, i.e., $W(T_n)=\pi(C(T_n)/C(\text{Sys}))$. The skilled artisan should recognize that other weight functions π can be utilized.

[0067] At block 712 the technique computes the expected value $L(n)$ of n by taking the subtree weight $W(T_n)$ and subtracting from it the weight of all subtrees rooted at its children. With a formula, this can be expressed as:

$$L(n)=W(T_n)-\sum_{\text{all children } m \text{ of } n} W(T_m).$$

If a node has no children, its expected value $L(n)$ is simply its weight, that is, $L(n)=W(n)$.

Note that due to the post-order traversal, all values $W(T_m)$ are already known to the algorithm. In one possible implementation of the technique, the values $L(n)$ are then interpreted as winning probabilities of each node n , and a lottree winner is probabilistically selected according to these probabilities.

[0068] At 714 the technique queries whether n_{cur} is the root of the solicitation tree. In an instance where n_{cur} is not the root (i.e., "no" at block 714), then the process proceeds to block 716 to continue the post-order traversal. In an instance where the n_{cur} is the root (i.e., "yes" at block 714) then the technique proceeds to block 718. In this case, reaching the root indicates that the process is complete due to the nature of the post-order traversal.

[0069] At block 716, the technique proceeds by letting n_{cur} be the next node in the post-order traversal of T_{sys} . From block 716, the technique returns to block 708. Accordingly, blocks 708-716 can be repeated until the traversal of the entire tree is complete.

[0070] At block 718, once all expected values $L(n)$ in the solicitation tree are computed (i.e., "yes" at 714), the winner is selected randomly in proportion to these expected values. That is, a node n with expected value $L(n)$ has a probability of $L(n)$ to win the lottery. The skilled artisan will realize that the win probability need not exactly match the expected value, because the payout value of the lottery can be varied; the combination of payout value and win probability together determines the expected value. This random selection can be done using a simple pseudo-random generator, for instance. Because Pachira's winner-selection mechanism requires only a single bottom-up traversal of the tree, its running time is linear in the number of participating nodes. Computational complexity is thus not a significant impediment to practical use of a Pachira lottree.

[0071] FIGS. 8-11 collectively present an example of how to compute the expected values $L(n)$ of a solicitation tree 800 in the Pachira scheme. Solicitation tree 800 as well as other lottery tree variants can be generated by a lottery tree mechanism (designated with specificity in relation to FIG. 5). The lottery tree mechanism can utilize the generated tree in selecting a lottery winner(s) for the associated network system. Recall the solicitation tree (with contributions) from the previous example depicting the Luxor scheme. As already pointed out, in this example the total contribution in this tree is $C(\text{Sys})=40$. This example further uses the parameters $\beta=0.6$ and $\delta=0.5$.

[0072] In this example respective individual nodes are shown with their contribution and expected value (represented as “contribution: expected value”). For instance, the first node to be computed is designated generally at **802** and has a contribution of 12. This node’s weight is $\pi(12/40)=0.246$ and because this node has no children, this is also its expected value $L(n)$. So in this case, the node is designated with “12: 0.246”. The next node to be considered in the post-order traversal is the left-most leaf-node with contribution 2 designated generally at **804**. This node’s weight is $\pi(2/40)=0.034$ and again because it does not have any children of its own, this also corresponds to this node’s expected value. The same also holds for its two siblings designated at **806, 808** that also have a contribution of 2. These two nodes are the next to be considered in the post-order traversal. Finally, the fifth step of the post-order traversal considers the node with contribution 8 designated at **810**.

[0073] Doing the same calculation as above reveals that this node **810** has an expected value of $L(n)=\pi(8/40)=0.156$. FIG. 8 shows the situation after these first five steps of the post-order traversal. (Grey nodes have already been computed.)

[0074] FIG. 9 shows the next node **902** to be considered is the inner node with contribution 2 that has four children (**804, 806, 808, and 810** with respective contributions 2,2,2,8). The weight of this node’s subtree is $W(T_n)=\pi((2+2+2+2+8)/40)=\pi(16/40)=0.341$. In order to compute the expected value of this participant, the technique subtracts the weight of all its children’s subtrees from this value. In this case, this yields

$$L(n)=0.341-0.034-0.034-0.034-0.156=0.083.$$

[0075] FIG. 9 shows the situation after the computation described above in relation to node **902**. Notice that although this participant’s contribution is 2 like three of its children, its expected value is higher, because of the “bonus” received for soliciting four children. This shows how Pachira manages to encourage both participation and solicitation.

[0076] FIG. 10 illustrates the next step that considers node **1002**. This node **1002** has a contribution 3 and has two children (**802, 902**), one (i.e., node **802**) with contribution 12 the other (i.e., node **902**) with contribution 2. The weight of this node’s subtree is $W(T_n)=\pi(3/40)=0.738$. Again, the technique can use this weight to compute the expected value of node **1002** by subtracting the weights of all its children’s subtrees, which have already been computed before. In this case, the values are:

$$L(n)=0.738-0.246-0.341=0.151.$$

[0077] FIG. 11 shows the situation after the computation described above in relation to FIG. 10. In this case, the next node will be the leaf node **1102** with contribution 1, followed by the leaf node **1104** with contribution 3, then its parent node **1106** with contribution 5, and finally, the root **1108** is considered. The computation follows the same proceedings described above and the final outcome of this computation can be seen in FIG. 11 with nodes **1102, 1104, 1106, and 1108** having expected values of 0.017, 0.053, 0.103, and 0.090, respectively.

[0078] As the final step, a winner is selected according to these expected values. For example, a node with expected value X may have a probability of winning X. Accordingly, the odds of an individual node being selected as the lottery winner relate to the node’s expected value. Under some circumstances, both the odds and the payout amount may be adjusted in a manner that keeps the expected value unchanged. In particular, it may be desirable for the root of the

tree never to be selected as the winner, since the root may represent the executive who is running the lottree. Yet, the root might have some expected value, as it does in the example of FIG. 11, wherein the root **1108** has an expected value of 0.090. In such a case, the root’s win probability can be distributed among the other nodes in proportion to their win probabilities, and the payout amount can be resealed to keep the expected value to all other nodes unchanged. This implementation offers an example of an exemplary technique that can reward both solicitation and contribution while discouraging or disincentivizing individual’s from gaming the system.

CONCLUSION

[0079] The above discussion relates to motivating participation in networked systems via a lottery payout. In some of the above implementations the odds of an individual participant in the network system being selected as the lottery winner depends upon the individual’s contribution and solicitation to the system. Although techniques, methods, devices, systems, etc., relating to motivating participation in networked systems are described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the claimed methods, devices, systems, etc.

1. A method, comprising:

gathering a solicitation hierarchy in a distributed system; tracking contribution to the distributed system of participants within the hierarchy; and, probabilistically determining a participant as a lottery winner based at least in part on the solicitation hierarchy and the contribution.

2. The method of claim 1, wherein the gathering comprises receiving data that uniquely identifies a new participant to the distributed system and an individual participant that solicited the new participant.

3. The method of claim 1, wherein the gathering comprises constructing a solicitation tree with participants represented as nodes of the solicitation tree.

4. The method of claim 3, wherein the tracking comprises computing an overall contribution for the solicitation tree and then computing a post-order traversal of the solicitation tree to weight the contribution of individual nodes.

5. The method of claim 3, wherein the tracking comprises resealing the solicitation hierarchy by redistributing a probability of winning among at least some of the participants.

6. The method of claim 1, wherein the probabilistically determining comprises addressing attempts by a potential participant to game the distributed system in an attempt to increase a probability that the potential participant will be selected as the lottery winner.

7. The method of claim 1, wherein the probabilistically determining comprises employing a schema that addresses attempts by a potential participant to increase a probability of the potential participant being selected as the lottery winner by joining the distributed system as multiple identities.

8. A network system comprising a lottery tree mechanism to encourage one or more of: network participation and network solicitation.

9. The network system of claim 8, wherein the lottery tree mechanism is configured so that odds of an individual net-

work participant being selected as a lottery winner are enhanced by both the individual's network participation and the individual's network solicitation.

10. The network system of claim 8, wherein the lottery tree mechanism is configured to employ a proportional selection lottery tree that does not include a solicitation structure.

11. The network system of claim 8, further comprising a mechanism for disincentivizing a participant from attempting to game the system to increase a likelihood of being selected as a winner of the lottery.

12. The network system of claim 11, wherein the mechanism for disincentivizing addresses both how and where the participant joins the network.

13. The network system of claim 8, wherein the lottery tree mechanism is configured to construct a hierarchical tree of network participants and to probabilistically select as a winner an ancestor of a chosen candidate winner in the tree.

14. The network system of claim 8, wherein the lottery tree mechanism is configured to construct a hierarchical tree of network participants and to generate a relative weighted winning probability for individual participants based upon any sub-trees of the individual participant.

15. A lottery tree mechanism configured to probabilistically encourage participants to both contribute to a network system and to solicit new participants to the networked system.

16. The lottery tree mechanism of claim 15, wherein the lottery tree mechanism is configured to track participants and to compute probabilities of winning for the participants based upon relative contribution and solicitation of individual participants.

17. The lottery tree mechanism of claim 15, wherein the lottery tree mechanism is configured to track participants as nodes of a lottery tree and to compute probabilities of individual participants being selected as a lottery winner through a post-order traversal of the lottery tree.

18. The lottery tree mechanism of claim 15, wherein the lottery tree mechanism is configured to utilize an algorithm for selecting a lottery winner that prevents an individual participant from having a higher probability of being selected as the lottery winner by splitting the individual's identity.

19. The lottery tree mechanism of claim 15, wherein the lottery tree mechanism is configured to utilize an algorithm that computes an individual's contribution in a manner that disincentivizes the individual from joining the network system multiple times.

20. The lottery tree mechanism of claim 15, wherein the lottery tree mechanism is configured to utilize an algorithm that computes an individual's contribution in a manner that disincentivizes the individual from gaming where the individual joins the network system.

* * * * *